11-24-00

A

PTO/SB/05 (08/00)
Approved for use through 10/31/2002, OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box [+]

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

JC871 U.S. PTO 11/22/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket	390086.94529
First Inventor	Joseph P. Debbins
Title	APPLICATION DEVELOPMENT SYSTEM FOR A MEDICAL IMAGING SYSTEM
Express Mail Label No.	EK 950 629 442 US

JC871 U.S. PTO 09/721221 11/22/00

APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents

ADDRESS TO: **Commissioner for Patents**
Box Patent Application
Washington, D.C. 20231

1. [X] Fee transmittal Form (Submit an original and a duplicate for fee processing)
2. [ ] Applicant claims small entity status See 37 CFR 1.27.
3. [X] Specification [Total Pages 26] *(preferred arrangement set forth below)*
   - Descriptive title of the invention
   - Cross References to Related Applications
   - Statement Regarding Fed Sponsored R&D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. [X] Drawing(s) (35 USC 113) [Total Sheets 7]
5. Oath or Declaration [Total Pages 3]
   a. [ ] Newly executed (original or copy)
   b. [ ] Copy from prior Application (37 CFR 1.63(d)) (for continuation/divisional with Box 17 completed)
      i. [ ] DELETION OF INVENTOR(S) Signed Statement attached deleting inventor(s) named in prior application, see 37CFR 1.63(d)(2) and 1.33(b).
6. [ ] Application Data Sheet. See 37 CFR 1.76
7. [ ] CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
   a. [ ] Computer Readable Form (CRF)
   b. *Specification Sequence Listing on*
      [ ] CD-ROM or CD-R (2 Copies); or
      [ ] paper
   c. [ ] Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. [ ] Assignment Papers (cover sheet & documents)
10. [ ] 37 CFR 3.73(b) Statement [ ] Power of Attorney (where there is an assignee)
11. [ ] English Translation Document *(if applicable)*
12. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
13. [ ] Preliminary Amendment
14. [X] Return receipt postcard (MPEP 503) *(Should be specifically itemized)*
15. [ ] Certified copy of priority Document(s) *(if foreign priority is claimed)*
16. [ ] Other:

17. **If a CONTINUING APPLICATION,** check appropriate box and supply the requisite information: and in a preliminary amendment or in an Application Data Sheet under 37 CFR 1.76

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application no. /

Prior application information: Examiner: Group/Art Unit:

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

18. CORRESPONDENCE ADDRESS

[ ] Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or [X] Correspondence address below

NAME	Terri S. Flynn				
ADDRESS	Quarles and Brady LLP				
	411 East Wisconsin Avenue				
CITY	Milwaukee	STATE	WI	ZIP CODE	53202
COUNTRY	USA	TELEPHONE	414-277-5000	FAX	414-271-3552

Name	Terri S. Flynn	Registration No. (attorney/Agent)	41,756
Signature	Terri S. Flynn	Date:	11/22/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

EXPRESS MAIL LABEL NO. EK 950 629 442 US

PTO/SB/17 (8/00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2001

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT | $710.00

Complete if Known	
Application Number	
Filing Date	Herewith
First Named Inventor	Josef P. Debbins
Group Art Unit	
Examiner Name	
Attorney Docket Number	390086.94529

METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 17-0055

Deposit Account Name: Quarles & Brady LLP

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Applicant claims small entity status See 37 CFR 1.27

2. Payment Enclosed:

☐ Check ☐ Credit Card ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	710	201	355	Utility filing fee	710.00
106	320	206	160	Design filing fee	
107	490	207	245	Plant filing fee	
108	710	208	355	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($)710.00

2. CLAIMS

		Extra		Fee from below		Fee Paid
Total Claims	19	-20**=	0	X 18.00	=	0.00
Independent	2	-3**=	0	X 80.00	=	0.00
Multiple Dependent Claims				270.00	=	0.00

** or number previously paid, if greater, For reissues see below

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	80	202	40	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim
109	80	209	40	**Reissue independent claims over original patent
110	18	210	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)0.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920	112	920	Requesting publication of SIR prior to Examiner action	
113	1,840	113	1,840	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	390	216	195	Extension for reply within second month	
117	890	217	445	Extension for reply within third month	
118	1,390	218	695	Extension for reply within fourth month	
128	1,890	228	945	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive unavoidably abandoned application	
141	1,240	241	620	Petition to revive unintentionally abandoned application	
142	1,240	242	620	Utility issue fee (or reissue)	
143	440	243	220	Design issue fee	
144	600	244	300	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	710	246	355	Filing a submission after final rejection (37 CFR 1.129(a))	
149	710	2496	355	For each additional invention to be examined (37 CFR 1.129(b))	
179	710	270	355	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) _____

* Reduced by Basic Filing Fee Paid SUBTOTAL (3) ($).00

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Terri S. Flynn	Registration No. (Attorney/Agent)	41,756	Telephone	414-277-5000
Signature				Date	11/22/01

APPLICATION DEVELOPMENT SYSTEM FOR A MEDICAL IMAGING SYSTEM

BACKGROUND OF THE INVENTION

The field of the invention is medical imaging systems, and particularly, systems for developing software applications for such imaging systems.

There are many types of medical imaging systems. The primary distinction between the different systems is the medical imaging modality that is used, such as, x-ray, magnetic resonance, ultrasound or nuclear. In addition, a broad range of capabilities and features are typically offered in each imaging modality. For example, a magnetic resonance imaging ("MRI") system may be offered with a range of polarizing magnetic strengths and configurations and with a range of different optional features such as magnetic resonance angiography ("MRA"), cardiac imaging and functional magnetic resonance imaging ("fMRI").

Despite the many differences, medical imaging systems have a number of basic functions in common. All medical imaging systems include an operator interface which enables a particular image acquisition to be prescribed, a data acquisition apparatus which uses one of the imaging modalities to acquire data from the subject, an image reconstruction processor for reconstructing an image using acquired data, and storage apparatus for storing images and associated patient information. Typically, hardware is designed to carry out these functions and system software is designed and written for each hardware configuration.

A medical imaging system contains application programs which direct the imaging system to perform particular types of scans, image reconstructions and post processing applications. For example, an MRI system may include application software which directs the imaging system

to perform a fast spin-echo scan, or a fast gradient-recalled echo scan, or a functional MRI scan, or a cardiac cine scan. Each of these different applications requires the writing of software code in a language such as assembler, or C and the linking and compiling of such code for use in the MRI system. As the number of applications grows, the amount and complexity of the application software code becomes increasingly difficult to maintain. As a result, the addition of new applications to the imaging system becomes increasingly difficult.

SUMMARY OF THE INVENTNION

The present invention is an application development system for a medical imaging system, and particularly, a system for producing an object oriented application program from a library of stored components, each component containing methods in the form of executable code and data related to the operation of the medical imaging system. The application development system includes a memory for storing a library of components; a display containing a framework area providing an indication of the components stored in the library, containing a workspace area for indicating components selected from the library to form an application program, and containing a properties area for indicating instance variables of a selected component; an input device; and a processor programmed to enable a user to select with the input device components indicated in the framework area and place them in the workspace area, programmed to edit with the input device instance variables indicated in the properties area for a selected component, and programmed to store the components in the workspace area as an application program.

Application programs may be created without writing and compiling program code. All program code is contained in the components stored in the library and the selection of components and placement in the workspace area of the display links the program code and any modified

properties, or instance variables into a single application program which can be "persisted". The application program is stored on the medical imaging system along side the component library, and when the application is selected to perform a scan, the program code from the proper components in the component library are linked together per the design of the stored application, and their instance variables set as indicated by the application program.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an MRI system which employs the preferred embodiment of the invention;

Fig. 2 is a block diagram of functional components in a workstation which forms part of the MRI system of Fig. 1;

Fig. 3 is a block diagram of the software architecture of a preferred embodiment of an application development system which employs the present invention;

Fig. 4 is a pictorial representation of a display produced by a visual component assembler which forms part of the software architecture of Fig. 3;

Fig. 5 is a block diagram of an imaging system which employs the present invention;

Fig. 6 is a pictorial display of an alternative embodiment of a workspace region in the display of Fig. 4;

Fig. 7 is a pictorial display of a properties portion of the display in Fig. 4 depicting an rf pulse component; and

Fig. 8 is a pictorial display showing a graphic plot of the rf pulse component of Fig. 7.

GENERAL DESCRIPTION OF THE INVENTION

Referring particularly to Fig. 5, a medical imaging system includes imaging apparatus 110 comprised of mechanical and electrical hardware elements that are operated during a scan to acquire image data. The imaging system also includes data processing apparatus 112 that is operated to reconstruct images using the acquired image data. To operate the system and to enter a scan prescription an operator input device 114, such as a keyboard and control panel, is provided, a display device 116 is provided to present the images for visualization and a storage device 117, such as a hard disc drive, is provided to archive acquired images. The particular imaging modality used, and the complexity and power of these hardware elements varies substantially from one system to the next.

The system includes a workstation 118 which is programmed in a machine independent language, such as Java™, to provide a user interface 120 that enables an operator to enter scan parameters using the operator input device 114. The workstation 118 is programmed to produce a scan description 122, which in its simplest configuration contains image acquisition description components and data processing description components that contain information required by the imaging apparatus 110 and data processing apparatus 112 to perform the prescribed scan.

Prior to run time, a snap shot of the scan description 122 is downloaded to a plurality of servers which control the imaging system hardware apparatus. In the simplest configuration these include an image acquisition server 124 and a data processing server 126 which operate the respective imaging apparatus 110 and data processing apparatus 112. When provided with the scan description components, the servers'

programs direct the image system hardware apparatus to perform the prescribed scan. A data store server 113 directs the storage device 117 to save the images along with associated patient information.

The particular scan or operation that is performed by the medical imaging system is directed by an application program stored in the workstation 118.

The application program is produced using an application development system that runs on the workstation 118 or a separate workstation (not shown). The application development system enables the user to create a new application program by selecting objects, or components, written in an object-oriented programming language, from a component library, and assemble them using a visual component assembler. The instance variables of selected components are displayed and may be edited for the new application program. The assembled components are instantiated and saved as a new application program which may be reconstituted for use on the medical imaging system. Instantiation is achieved using a serialization process in which the hierarchical relationship of components and their instance variables are stored.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring particularly to Fig. 1, the preferred embodiment of the invention is employed to operate an MRI system. The MRI system includes a workstation 10 having a display 12 and a keyboard 14. The workstation 10 includes a processor 16 which is a programmable machine commercially available from Silicon Graphics, Inc. It is based on a 64-bit microprocessor manufactured by Intel and it runs the Linux operating system. The workstation 10 provides the operator interface which enables scan prescriptions to be entered into the MRI system. As will be described

in more detail below, the workstation 10 will run one or more Java™ virtual machines which will run code which is programmed in the Java™ language that is fully transportable to any other programmable machine which is Java™ compatible.

The workstation 10 is coupled to four servers: a pulse sequence server 18; a data acquisition server 20; a data processing server 22, and a data store server 23. In the preferred embodiment the data store server 23 is performed by the workstation processor 16 and associated disc drive interface circuitry. The remaining three servers 18, 20 and 22 are performed by separate processors mounted in a single enclosure and interconnected using a 64-bit backplane bus structure based on the PCI standard for industrial and telecommunications applications called "CompactPCI". The pulse sequence server 18 employs a 366 MHz microprocessor model PPC750 and a quad communication controller model MPC860T manufactured by Motorola, Inc. The data acquisition server 20 and data processing server 22 both employ the same 366 MHz microprocessor and the data processing server 22 further includes one or more array processors based on parallel vector processors commercially available from Mercury Computer Systems, Inc. as the PowerPC™. Another 366 MHz microprocessor (not shown) serves as a hardware controller on the PCI bus structure and it controls a quad communication controller model MPC860T manufactured by Motorola, Inc.

The workstation 10 and each processor for the servers 18, 20 and 22 are connected to a 100 BaseT Ethernet serial communications network. This serial network conveys data that is downloaded to the servers 18, 20 and 22 from the workstation 10 and it conveys tag data that is communicated between the servers and between the workstation and the servers. In addition, a high speed data link using the BIT3 protocol is

provided between the data processing server 22 and the workstation 10 in order to convey image data to the data store server 23.

The pulse sequence server 18 functions in response to program elements downloaded from the workstation 10 to operate a gradient system 24 and an RF system 26. Gradient waveforms necessary to perform the prescribed scan are produced and applied to the gradient system 24 which excites gradient coils in an assembly 28 to produce the magnetic field gradients G_x, G_y and G_z used for position encoding NMR signals. The gradient coil assembly 28 forms part of a magnet assembly 30 which includes a polarizing magnet 32 and a whole-body RF coil 34.

RF excitation waveforms are applied to the RF coil 34 by the RF system 26 to perform the prescribed magnetic resonance sequence. Responsive NMR signals detected by the RF coil 34 are received by the RF system 26, amplified, demodulated, filtered and digitized under direction of commands produced by the pulse sequence server 18. Exemplary RF systems are described in U.S. Pat. No. 4,952,877 and U.S. Pat. No. 4,992,736.

The pulse sequence server 18 also optionally receives patient data from a physiological acquisition controller 36. The controller 36 receives signals from a number of different sensors connected to the patient, such as ECG signals from electrodes or respiratory signals from a bellows. Such signals are typically used by the pulse sequence server 18 to synchronize the performance of the scan.

The pulse sequence server 18 also connects to a scan room interface circuit 38 which receives signals from various sensors associated with the condition of the patient and the magnet system. It is also through the scan room interface circuit 38 that a patient positioning system 40

receives commands to move the patient to desired positions during the scan.

It should be apparent that the pulse sequence server 18 performs real-time control of MRI system elements during a scan. As a result, it is necessary that its hardware elements be operated with program instructions that are executed in a timely manner. As will be explained in more detail below, the pulse sequence server 18 is controlled during run-time by programs written in a low level programming language such as assembler, C or C++. The description components for a scan prescription are downloaded from the workstation 10 in the form of objects. The pulse sequence server 18 contains programs which receive these objects using a serialization mechanism. The pulse sequence server 18 also includes a program which converts the objects to C++ objects that are employed by the run-time programs. In the preferred embodiment Java™ objects are downloaded and the Java™ serialization mechanism is employed. The pulse sequence server 18 thus includes both hardware independent programs written in Java™ and hardware dependent programs. It is contemplated that Java™ interpreters will eventually become fast enough that nearly all programs run on the pulse sequence server 18 will be written in hardware independent form.

The digitized NMR signal samples produced by the RF system 26 are received by the data acquisition server 20. The data acquisition server 20 operates in response to description components downloaded from the workstation 10 to receive the real-time NMR data and provide buffer storage such that no data is lost by data overrun. In some scans the data acquisition server 20 does little more than pass the acquired NMR data to the data processor server 22. However, in scans which require information derived from acquired NMR data to control the further performance of the scan, the data acquisition server 20 is programmed to produce such

information and convey it to the pulse sequence server 18. For example, during prescans NMR data is acquired and used to calibrate the pulse sequence performed by the pulse sequence server 18. Navigator signals may be acquired during a scan and used to adjust RF or gradient system operating parameters or to control the view order in which k-space is sampled. And, the data acquisition server 20 may be employed to process NMR signals used to detect the arrival of contrast agent in an MRA scan as described in co-pending U.S. Pat. Appln. Serial No. 08/635,078 filed April 19, 1996 and entitled "*Method For Performing Magnetic Resonance Angiography Using a Contrast Agent*". In all these examples the data acquisition server 20 acquires NMR data and processes it in real-time to produce information which is used to control the scan.

As with the pulse sequence server 18, the hardware elements of the data acquisition server 20 are operated at run-time with program instructions in a programming language such as assembler, C or C++. As will be explained in more detail below, the directions for its operation during a scan are downloaded from the workstation 10 in the form of objects. A server proxy receives the objects using the serialization mechanism and the downloaded objects are converted to C++ objects that are employed to operate the data acquisition server 20 during run-time. As indicated above, Java™ objects are downloaded in the preferred embodiment using the Java™ serialization mechanism.

The data processing server 22 receives NMR data from the data acquisition server 20 and processes it in accordance with description components downloaded from the workstation 10. Such processing may include, for example: Fourier transformation of raw k-space NMR data to produce two or three-dimensional images; the application of filters to a reconstructed image; the performance of a backprojection image

reconstruction of acquired NMR data; the calculation of functional MR images; the calculation of motion or flow images, etc.

Images reconstructed by the data processing server 22 are conveyed back to the workstation 10 where they are stored. Real-time images are stored in a data base memory cache (not shown) from which they may be output to operator display 12 or a display 42 which is located near the magnet assembly 30 for use by attending physicians. Batch mode images or selected real time images are stored in a host database on disc storage 44. When such images have been reconstructed and transferred to storage, the data processing server 22 notifies the data store server 23 on the workstation 10. The workstation 10 may be used by an operator to archive the images, produce films, or send the images via a network to other facilities.

Directions for the particular operations to be performed by the data processing server 22 are downloaded from the workstation 10. The time critical functions are performed with programs written in assembler, C or C++ and the downloaded Java™ object directions must be converted to corresponding executable code as described above.

As indicated above, the workstation 10 is a Java™ virtual machine which executes programs written in the Java™ programming language. The workstation software is structured to perform "applications" which may be selected and run by an operator. Such applications correspond to clinical imaging procedures and may include, for example:

perform a scan using an FSE pulse sequence;
conduct a CEMRA dynamic study;
perform an fMRI study;
perform a runoff vascular study
perform image post processing

filming
networking

An application is a collection of Java™ objects stored in an "application container" that may be selected by an operator to perform a scan. Referring particularly to Fig. 2, each application container includes a Java™ application controller component 46 which directs other Java™ components in the container to perform the scan. These other components include a prescription controller 52 which includes a user interface component 53 and a prescription assistant component 55 that enable an operator to control the procedure performed by the application.

The application container also includes scan descriptions 50. These scan descriptions are downloaded to the servers 18, 20, 22 and 23 (Fig. 1) and used by those servers to perform the prescribed scan. The stored scan descriptions 50 are unique for every different application.

The preferred embodiment of the present invention is an application development system which produces application programs for this MRI system. This application program is a collection of interrelated Java™ objects within the application container Java™ object. These objects are selected and edited using tools in the application development system. The application development system may reside on the MRI system workstation 10, or it may reside on a separate, stand alone workstation of similar structure and capability.

After the application program is developed, the application container object is serialized and stored in the disc storage 44. When the operator of the MRI system selects the application program, the corresponding serialized application container object is read from the disc memory 44 and reconstituted as shown in Fig. 2 to operate the MRI system. To better understand the requirements of the application development system, the

operations performed by the MRI system under the direction of the application program will now be described.

The application controller 46 includes an application state object 48 which maintains the state of the application as the scan is performed. The possible states during a life cycle of an application are as follows:

Initialization
Prescribing
Prescribed
Downloading
Downloaded
Prescanning
Prescanned
Batch Scanning
Real Time Scanning
Scan Paused
Scanned
Reconstructed
Visualized.

This life cycle is driven by commands from the application container (like initialize application), by commands from the operator (like start scan) and by commands generated internally by the application (like scan done).

When the operator initially selects the application, the application initializes and changes to the "prescribing state" and the prescription controller 52 is enabled to interact with the scan description components 50 to determine what scan parameters must be specified by the operator (e.g. TR, number of slices, location of FOV, flip angle) and determine if the prescription is complete and valid. The prescription controller 52 then

signals the application state object 48 to switch to the "prescribed" state and download, prescan and scan buttons on the control panel are enabled.

If the operator hits the "download" button, the application state object 48 changes to the "download state" and the application controller 46 employs a snap shot controller 54 to issue snap shot and download commands. As will be described in more detail below, these commands cause the scan descriptions 50 to be downloaded to the servers 18, 20, 22 and 23. The snap shot controller 54 receives "download done" notification back from each of the servers 18, 20, 22 and 23, and when all four servers have been downloaded, the application state object 48 is changed to the "downloaded" state.

If the operator hits the scan button, the application state object 48 will change to the scan mode and a scan controller 56 is employed to issue a scan command to the pulse sequence server 18. The next state transition is governed by the scanning mode i.e., real-time or batch. The behavior of the application in the two modes is very different and so there are two different scanning states. If in real-time mode, the application is set to a "real-time scanning" state and if in batch mode, the application state is set to a "batch scanning" state. When in the real-time mode, if the user chooses to pause the scan, the application will transition to a "scan paused" state. If scanning is resumed, the application goes back to the real-time scanning state. In real-time scanning state, the application can be edited and edited descriptions will be downloaded even while the scanning is in progress. However, the application will not make a state transition; instead, the same state will be characterized to allow editing and downloading. It is this behavior of the real-time scanning state that differentiates it from the batch scanning state.

The application will make a transition to the "scanned" state when the operator hits the "stop scan" button. Also, if the application is in the

batch scanning mode of operation, the pulse sequence server 18 notifies the application controller 46 when the scan is completed. The application state object changes to the "scanned" state in either event.

When the data processing server 22 completes reconstruction of the acquired images, the application controller 46 is notified and the application state object 48 is changed to the "reconstructed" state. This indicates to the workstation 10 that reconstructed images are available on disk 44 for display or further processing.

The scan descriptions 50 contain a set of components that serve to collect scan parameters using the prescription controller 52, and to organize those prescription scan parameters into a set of smaller components that can be downloaded to the servers 18, 20, 22 and 23. On the servers 18, 20, 22 and 23, those downloaded components direct the operation of the hardware in order to carry out the prescribed scan.

There are different description types within each application to provide logical groupings of components to deal with different aspects of executing an MR scan. These description types are:

Pulse Description;
Sequence Description;
Acquisition Description;
Data Processing Description;
Data Store Description.

The pulse description includes components that define and control the waveforms to be played out on the gradient system and the RF system hardware, along with hardware control components. These components control the dynamic aspects of the waveforms and hardware in response to events produced at run-time by components of the sequence description. This description also includes components that control the filtering of NMR

signals received by the RF system 26. These components collectively define a unique set of gradient/RF/control pulses which are used to excite, encode, and readout the NMR signals. Examples are pulse descriptions for 2D spin echo, 2D gradient-echo, 2D fast spin-echo, and 3D gradient-echo sequences.

The sequence description includes a set of components that control the order of pulse sequences played out, and define a series of prescribed events along the scan timeline. These prescribed events defined by the sequence description trigger the dynamic behavior of the pulse components in the pulse description. These components prescribe a unique acquisition ordering used to define the slice and k-space sampling order. Examples are 2D sequential, 2D interleaved, 3D sequential, 3D elliptical centric, and multi-slice CINE.

The acquisition description includes a set of components that prescribe the real-time processing of NMR signals acquired by the RF system 26. These components direct the performance of operations on acquired NMR signals to produce information that is fed back to components in the sequence description to affect subsequent scanner operation. These components may, for example, process NMR signals during a calibration prescan to feedback changes in the power or frequency of RF pulses produced during the subsequent scan; or process NMR signals to detect when a bolus of contrast agent arrives in a region of interest and trigger the start of a centric view order acquisition; or process "navigator" NMR signals to produce phase correction information which may be used to alter the view order of the scan or alter the demodulation reference frequency of the RF system 26. There are scans commonly used in clinical applications which do not require this capability, however, and in those applications, the components in the acquisition description simply

buffer or filter the acquired NMR signals and make them available to the data processing server 22.

The data processing description contains components that direct the data processing server 22 to transform acquired NMR signals into a meaningful form. Image reconstruction is the most common function and the resulting form is a 2D or 3D image of the subject being scanned. Spectroscopy processing can also be defined by these components, in which case the form that results is an image of the spectra of the acquired NMR signals.

The data store description contains components that define the images which are stored in the database during a scan. In addition to the reconstructed images, this may include patient information and scan parameter information which is to appear annotated on the image along with the patient anatomic or spectrographic information.

As indicated above, the application development system is implemented on a workstation having a memory, a display, an input device such as a keyboard and mouse and a processor programmed to perform the functions now to be described. Referring particularly to Figs. 3 and 4, the programs and data which form the software architecture of the application development system includes a visual component assembler 60 which produces a window display 62 to the user that contains three areas: a framework area 64; a workspace area 66; and a properties area 68. The application program is developed by selecting components from the component library 72 which are displayed in the framework area 64 and dragging them into the workspace area 66. Such selected components are stored in workspace storage 67. The properties of a selected component in the workspace area 66 are displayed in the properties area 66, and these properties can be changed using an property editor 70.

The components displayed in the framework area 64 are Java™ classes, or objects stored in a component library 72. These components are typically developed using a commercially available integrated development environment 74 such as that sold under the trademark "Forte for Java" by Sun Microsystems or "JBuilder" sold by Inprise. The components are written in Java™ source code and compiled into binary instructions called byte code. These byte code components are saved to the appropriate packages in the component library 72. It can be appreciated that many Java components are commercially available and these can be used along with custom written Java components to create more complex components specifically applicable to performing the MRI functions described above. The objective is to create and store enough components in the component library 72 such that a user can build any desired application by selecting existing components. In such case, the user does not need to write any software code to implement new applications, but is simply requested to select and aggregate the desired functionality.

Referring particularly to Figs. 3, 4 and 6, as components are dragged into the workspace 66 from the framework area 64, the visual component assembler 60 establishes the hierarchical relationship between the components. These relationships are illustrated in one preferred manner in Fig. 4 by the indentation of containee component icons beneath their related container component icon. An alternative embodiment of the display of this hierarchical relationship of components in the workspace 66 is shown in Fig. 6. In this embodiment, arrows point from each superclass component icon to its related subclass component icons. It can be appreciated that the display of a complete application requires more display area than it available and that scroll bars may be used to display different portions of the application in the workspace 66.

To build an application, the user first loads the framework area 64 with the library of components. The components are displayed and an application container component 76 is selected. When this component 76 is dragged to the workspace 66, components for all of its children are also identified. This initiates the building of an application, but the user must know what further components are required to complete the build. To assist in this effort any component in the workspace 66 can be selected with a right click of the mouse and a description of the function performed by that component is displayed in the format known as Javadoc™.

Each component in the workspace 66 has properties, which include numeric variables, Boolean variables and other class type or named variables. The visual component assembler enables the user to display these properties in the properties area 68 by left clicking on a selected component. If the user then left clicks to select one of the displayed properties in the properties area 66, the property editor tool 70 is employed to enable the user to make the change. It is contemplated that most of the new applications created in clinical settings will be limited to changing the properties in components of existing applications. In other words, existing applications are dragged from the framework area 64; the properties in selected components therein are edited; and the result is saved back to the component library 72 as a new application.

When the application is completed the selected components assembled in the workspace area 66 are then persisted. This is currently accomplished by storing the application using the above described serialization mechanism, however, other persistence mechanisms are known and may be used. The persistence mechanism stores the hierarchical relationship ("graph") between the selected components as well as the instance variable (property) values. The byte code for each component in the application is not stored with the persisted application.

The medical imaging system which employs the application must itself store the byte code for all the components used in applications. When the persisted application is restored, or "deserialized", on the MRI system, it directs the loading of the byte code indicated by the persisted object graph and instance variable.

Because many of the components in MRI applications relate to the production of pulses and pulse sequences, another feature of the present invention is the display of waveforms or other data produced by a component. Referring particularly to Figs. 3, 7 and 8, all components, may include a property called a "visible." In the case of a pulse waveform component, when the visible property is switched to "true", a waveform plotter 78 is enabled and a pulse sequence plotter window 80 is produced and displayed. This window 80 displays the waveforms 82 and 84 which are produced by this instance of the component. If a property is changed using the editor 70, the displayed waveforms may also change. This is illustrated in Fig. 8 when the property "pulse type" is about to be changed.

When the application is complete, the application container object is serialized and saved to storage. As indicated above, this serialized form of the application program may be restarted in another virtual Java™ machine such as the MRI system described above and reconstituted into an object-oriented application program ready for execution.

One advantage of the serialized configuration of the stored application program is that it can efficiently be downloaded to clinical MRI systems from remote sites. This may be done through direct serial connection using a private Intranet or a public telephone system, or it may be done through the Internet public system. In any case, the transfer of the application program is a serialized object stream which carries the class name of each object, or component, as well as that component's instance data which is described by attribute name, type, and value. Also

transferred is the relationships between components which allows a graph of the components in the application program to be collected in the serial stream received at the MRI system and then recreated, or reconstituted, on the MRI system. The serialization mechanism follows all relationships between objects. Each object, or component in the graph is only serialized once. Should a component be referenced more than one time, the serialization process recognizes the repeat and inserts a reference to the previous occurrence in the graph. This prevents duplication of objects and reduces the magnitude of the download task.

The serialization process also eliminates the need for downloading byte code. This presumes that any MRI system that receives a downloaded application program has a library which stores the byte code for all components contained in the application program. Updates to clinical systems may, therefore, include both the downloading of new applications as well as the downloading of any necessary new components for the MRI system library.

Of course, the serialization process also enables applications to be uploaded from clinical MRI systems. This enables applications and/or components developed at clinical research systems to be uploaded to the MRI system manufacturer for review and analysis.

The application development system can also be operated in a simulation mode to try out the application program before an actual scan is performed on the MRI system hardware. This simulation capability is facilitated by the fact that the application program will run on any Java™ virtual machine with a suitable component library. It is also facilitated by the fact that the user interface is a component of the application, such that the simulation uses the same interface and displays the same information as when running on an MRI system. Simulation environment is a mode of operation of the application development system which allows a developer

to test and debug his/her application in a near-scanner like environment. The Java application loaded in the component assembler collection can be first saved and then simulation may be started. During simulation, the servers are in "simulation mode" and thus certain hardware interfaces are emulated or non-functional. Typically raw data is injected into the server imaging chain to be processed as it would if received by the transceiver.

During application simulation, the developer will have the opportunity to do certain levels of message tracing and component debug. The developer is provided with several levels of component message tracing, which can be set dynamically during application simulation. The developer may also invoke several levels of debug. During component and application development, the developer can set a "debug=TRUE" property in each component in order to access custom debug behavior for that component. Or, there may be several levels of debug for a property, with "0" being none or "debug off". For example, the sequencing component of the sequence description may provide for single stepping of the slice and phase encoding looping. Any custom user interfaces required for a given component are provided by that component in debug mode.

There may also be one or more simulation user interfaces which can be accessed during simulation to provide access to more global operations, such as observing the internal behavior of a specific server, inter-server tag communications or perhaps observing detailed behavior of a specific server, including tags and agent behavior.

CLAIMS

1. An application development system for a medical imaging system, which comprises:

a component library for storing components written in an object-oriented programming language; and

a visual component assembler for displaying in a framework area components in the component library and enabling a user to select components in the framework area and assemble them in a workspace area, and the visual component assembler being operable to persist components in the work area to form an application program for the medical imaging system.

2. The system as recited in claim 1 in which the visual component assembler also displays a properties area and enables a user to select a component in the framework area and display a set of properties associated with the selected component in the properties area.

3. The system as recited in claim 1 in which the persistence is performed by serializing components in the framework area.

4. The system as recited in claim 3 in which the serializing includes storing a hierarchical relationship between application components and storing their properties.

5. The system as recited in claim 2 which includes a property editor which enables a user to change the properties displayed in the properties area.

6. The system as recited in claim 5 in which one of the properties displayed in the properties area invokes a visual representation of the component, and the system includes means for displaying the visual representation.

7. The system as recited in claim 6 in which the visual representation is a waveform and the means is a waveform plotter.

8. The system as recited in claim 7 which includes a display having a screen on which the framework area, the workspace area and the properties area are displayed.

9. The system as recited in claim 8 in which the waveform plotter produces a window on the display screen in which the waveform appears.

10. The system as recited in claim 7 in which the property editor is operable to change the visual representation automatically when another property is changed.

11. The system as recited in claim 1 in which the object-oriented programming language is Java™.

12. The system as recited in claim 3 in which the means for persisting employs a Java™ object serialization mechanism.

13. A system for producing an application program for a magnetic resonance imaging system, which comprises:

a memory for storing a library comprising components written in an object-oriented programming language;

a workstation having a display, an input device and a processor programmed to perform application development functions, the application development program including:

a visual component assembler for displaying in a framework area on the display icons representing components in the component library and responsive to directions from a user entered through the input device to select components and assemble icons representative of the selected components in a workspace area displayed on the display; and for persisting the selected components to form an application program.

14. The system as recited in claim 13 in which the persistence is performed using a serialization mechanism which stores the application program.

15. The system as recited in claim 13 in which the visual component assembler also displays a properties area on the display and it enables a user to select a component and display properties associated with the selected component in the properties area.

16. The system as recited in claim 15 in which the application development program also includes a property editor which enables a user to input data through the input device to change property values displayed in the properties area.

17. The system as recited in claim 16 in which one of the properties displayed in the properties area is a visual representation of the component and the application development program also includes a waveform plotter for displaying the visual representation.

18. The system as recited in claim 17 in which the waveform plotter produces a window on the display in which the visual representation is produced.

19. The system as recited in claim 17 in which the property editor is operable to change the visual representation automatically when another property is changed.

Sandia National Laboratories

APPLICATION DEVELOPMENT SYSTEM FOR A MEDICAL IMAGING SYSTEM

ABSTRACT

A workstation is programmed to operate as an application development system for a medical imaging system. Objects programmed in an object-oriented language are selected from a component library using a visual component assembler which enables them to be dragged from a framework area on a display to a workspace area. Properties of selected components may be edited, and the resulting collection of components may be saved as an application program.

DO NOT COPY

09212233.112200

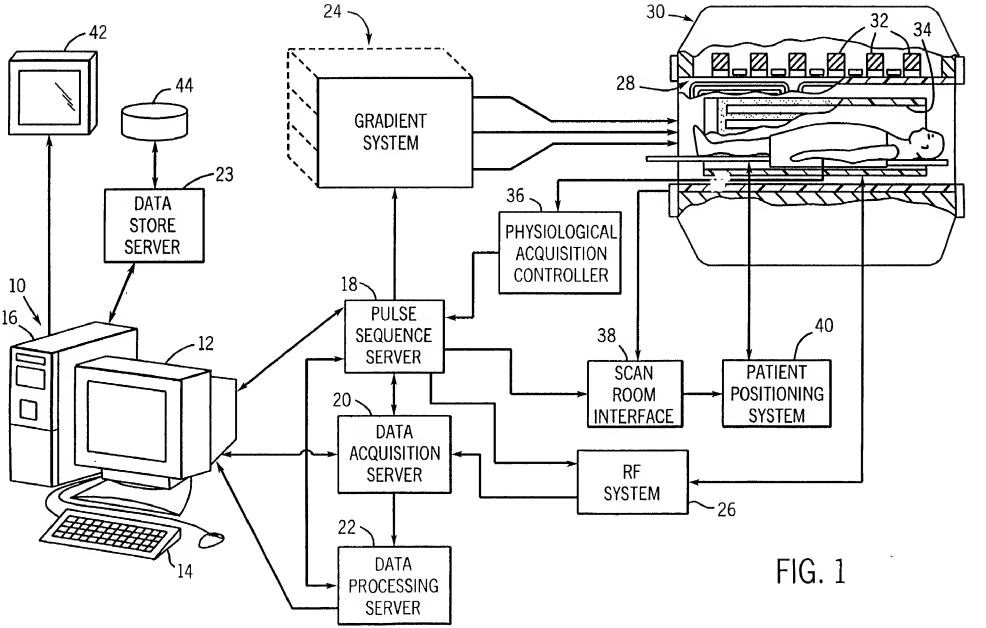


FIG. 1

FIG. 2

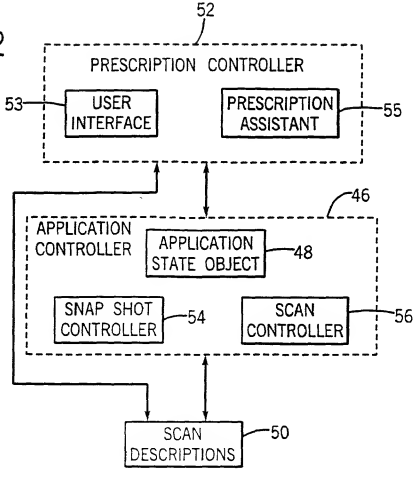
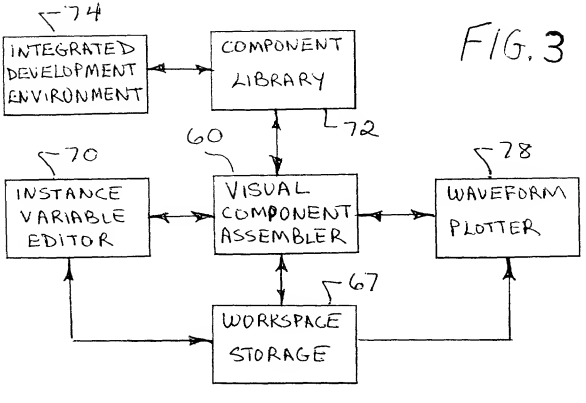

52
PRESCRIPTION CONTROLLER
53
USER INTERFACE
PRESCRIPTION ASSISTANT
55
46
APPLICATION CONTROLLER
APPLICATION STATE OBJECT
48
SNAP SHOT CONTROLLER
54
SCAN CONTROLLER
56
SCAN DESCRIPTIONS
50

FIG. 3


74
INTEGRATED DEVELOPMENT ENVIRONMENT
COMPONENT LIBRARY
72
60
70
INSTANCE VARIABLE EDITOR
VISUAL COMPONENT ASSEMBLER
78
WAVEFORM PLOTTER
67
WORKSPACE STORAGE

09/2211,2260

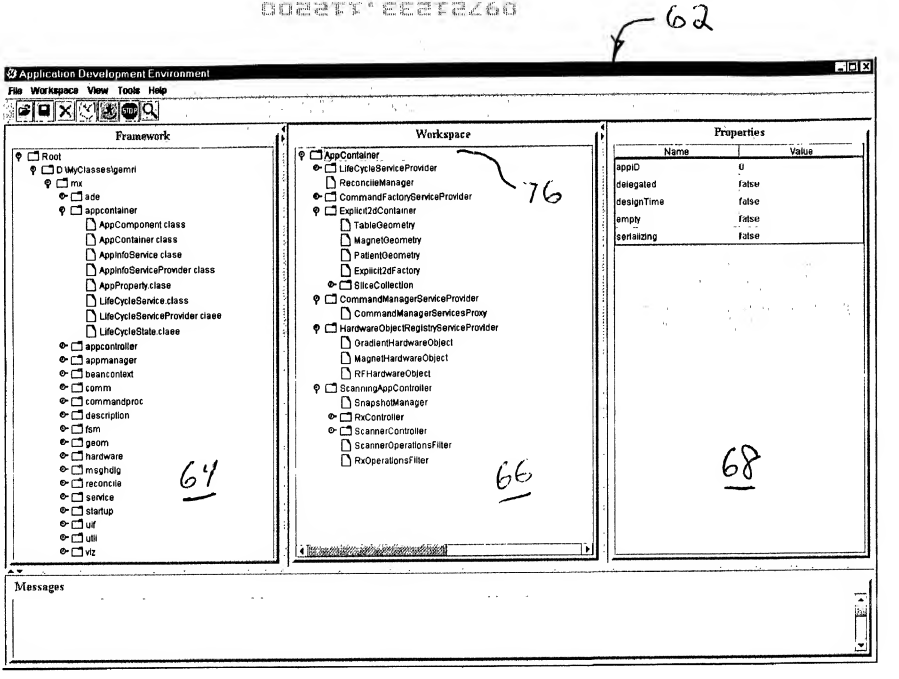
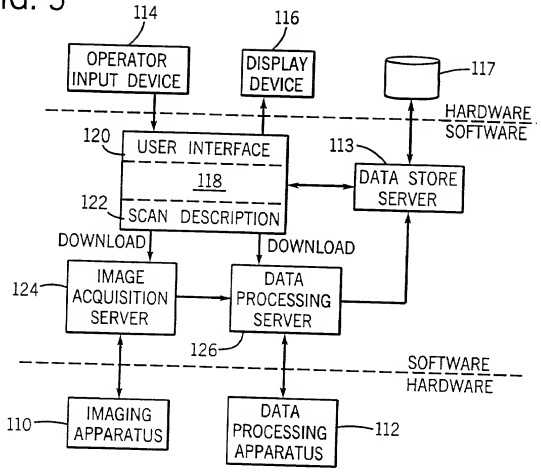


FIG. 4

FIG. 5


114
116
OPERATOR INPUT DEVICE
DISPLAY DEVICE
117
HARDWARE
SOFTWARE
120
USER INTERFACE
118
122
SCAN DESCRIPTION
113
DATA STORE SERVER
DOWNLOAD
DOWNLOAD
124
IMAGE ACQUISITION SERVER
DATA PROCESSING SERVER
126
SOFTWARE
HARDWARE
110
IMAGING APPARATUS
DATA PROCESSING APPARATUS
112

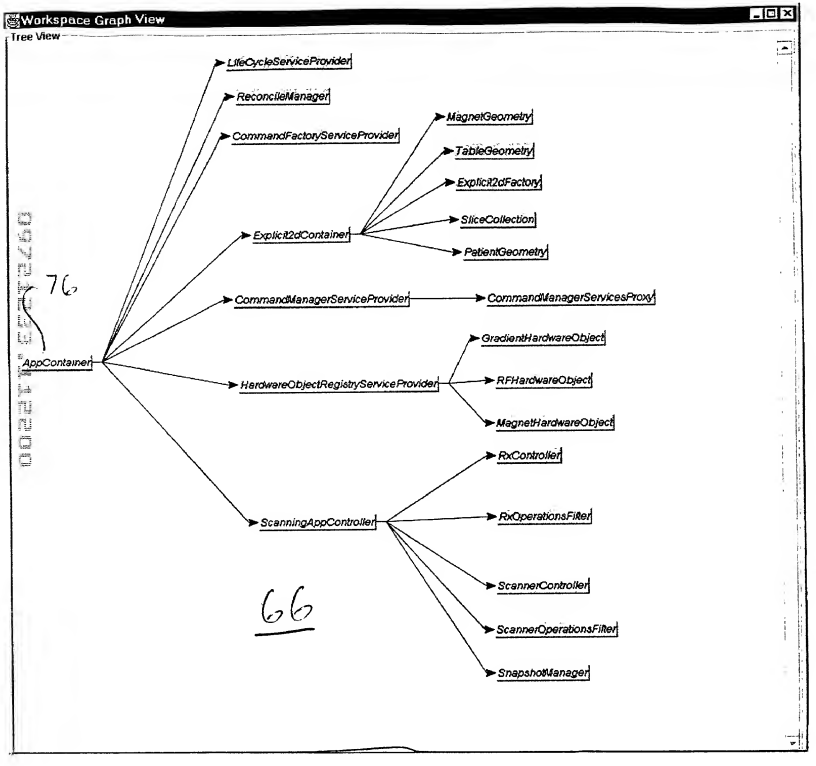

Workspace Graph View
Tree View
LifeCycleServiceProvider
ReconcileManager
CommandFactoryServiceProvider
MagnetGeometry
TableGeometry
Explicit2dFactory
SliceCollection
PatientGeometry
Explicit2dContainer
CommandManagerServiceProvider
CommandManagerServicesProxy
AppContainer
GradientHardwareObject
RFHardwareObject
MagnetHardwareObject
HardwareObjectRegistryServiceProvider
RxController
RxOperationsFilter
ScannerController
ScannerOperationsFilter
SnapshotManager
ScanningAppController
76
66

FIG. 6

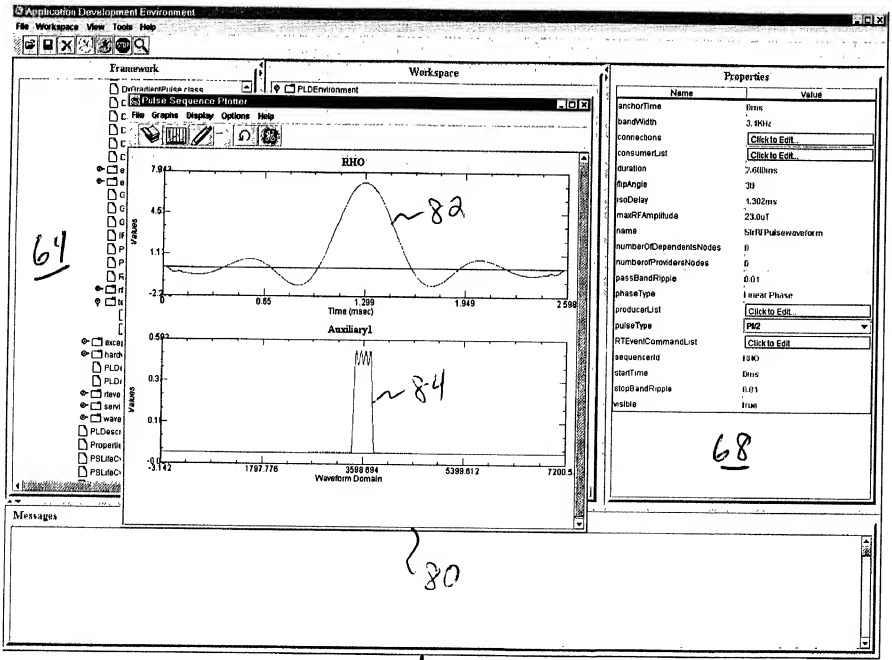

Application Development Environment
File Workspace View Tools Help
Framework
Workspace
PLDEnvironment
Pulse Sequence Plotter
File Graphs Display Options Help
RHO
82
Time (msec)
Auxiliary1
84
Waveform Domain
Properties
Name
Value
anchorTime
bandWidth
connections
Click to Edit...
consumerList
Click to Edit...
duration
flipAngle
30
fsoDelay
maxRFAmplitude
name
numberOfDependentsNodes
numberofProvidersNodes
passBandRipple
phaseType
Linear Phase
producerList
Click to Edit...
pulseType
PI2
RTEventCommandList
Click to Edit
sequenceId
startTime
stopBandRipple
visible
true
Messages
64
68
80
62

FIG. 7

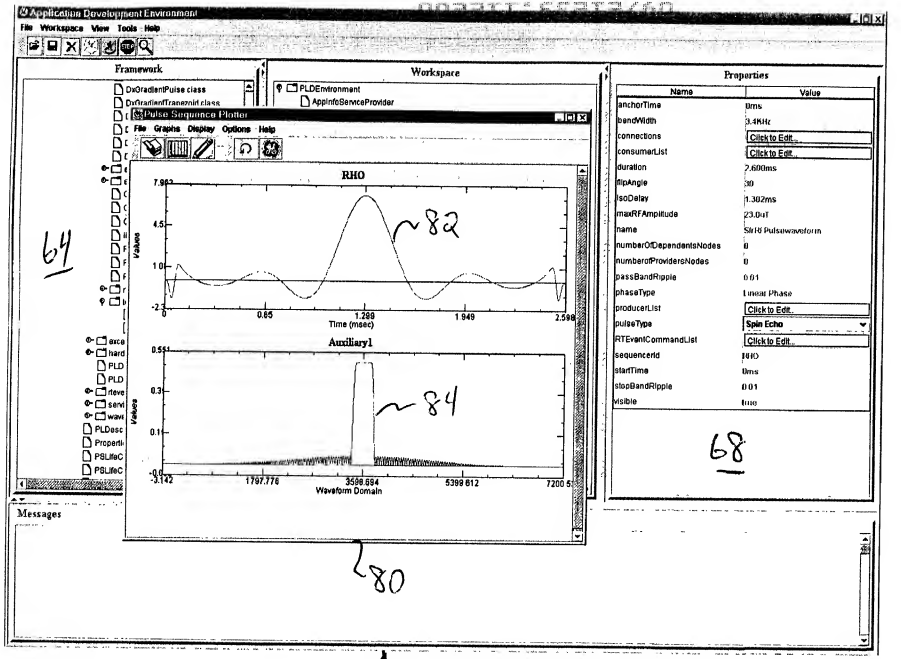

Application Development Environment
File Workspace View Tools Help
Framework
DxGradientPulse class
DxGradientTrapezoid class
Workspace
PLDEnvironment
AppInfoServiceProvider
Pulse Sequence Plotter
File Graphs Display Options Help
RHO
7.993
4.5
1.0
-2.2
Values
0
0.65
1.299
1.949
2.598
Time (msec)
82
Auxiliary1
0.51
0.3
0.1
-0.0
-3.142
1797.776
3598.694
5399.812
7200.5
Waveform Domain
84
Messages
Properties
Name
Value
anchorTime
0ms
bandWidth
9.4KHz
connections
Click to Edit...
consumerList
Click to Edit...
duration
2.600ms
flipAngle
90
isoDelay
1.302ms
maxRFAmplitude
23.0uT
name
Slr18 Pulsewaveform
numberOfDependantsNodes
0
numberofProvidersNodes
0
passBandRipple
0.01
phaseType
Linear Phase
producerList
Click to Edit...
pulseType
Spin Echo
RTEventCommandList
Click to Edit...
sequenceId
1000
startTime
0ms
stopBandRipple
0.01
visible
true
64
68
80
62

FIG. 8

EXPRESS MAIL LABEL NO. Ek 950 629 442 US

PTO/SB/01 (6-95)
Approved for use through 9/30/98. OMB 0651-0032
Please type a plus sign (+) inside this box ☐ Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

0010/PTO Rev. 6/95 — U.S. Department of Commerce Patent and Trademark Office

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION

☒ Declaration Submitted with Initial Filing OR ☐ Declaration Submitted after Initial Filing

Attorney Docket Number	390086.94529
First Named Inventor	Josef P. Debbins
COMPLETE IF KNOWN	
Application Number	
Filing Date	Herewith
Group Art Unit	
Examiner Name	

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

APPLICATION DEVELOPMENT SYSTEM FOR A MEDICAL IMAGING SYSTEM

(Title of the Invention)

the specification of which

☒ is attached hereto

OR

☐ was filed on (MM/DD/YYYY) ☐☐☐☐☐☐☐☐ as United States Application Number or PCT International

Application Number ☐☐☐☐☐☐☐☐ and was amended on (MM/DD/YYYY) ☐☐☐☐☐☐☐☐ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code §119(a)-(d) or §365(b) of any foreign application(s) for patent or inventor's certificate or §365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

☐ Additional foreign applications numbers are listed on a supplemental priority sheet attached hereto:

I hereby claim the benefit under Title 35, United States Code §119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	
		☐ Additional provisional application numbers are listed on a supplemental priority sheet attached hereto.

Burden Hour Statement: This form is estimated to take .4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box [+]

DECLARATION

I hereby claim benefit under Title 35, United States Code §120 of any United States application(s), or §365(C) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application or PCT international application in the manner provided in the first paragraph of Title 35, United States Code §112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations §1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application Number	PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number *(if applicable)*

[ ] Additional U.S. or PCT international application numbers are listed on a supplemental priority sheet attached hereto

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and all continuation and divisional applications based thereon, and to transact all business in the Patent and Trademark Office connected therewith:

[ ] Firm Name _____ Customer Number or label _____

OR

[X] List attorney(s) and/or agent(s) name and registration number below

Name	Registration Number	Name	Registration Number
Herbert W. Mylius	24,578	Richard T. Roche	38,599
Barry E. Sammons	25,608	Alexander B. Ching	41,669
Charles W. Jirauch	26,196	Terri S. Flynn	41,756
Nicholas J. Seay	27,386	John T. Pienkos	42,997
George E. Haas	27,642	Daniel G. Radler	43,028
Michael J. McGovern	28,326	Gregory M. Smith	43,136
Carl R. Schwartz	29,437	Steven J. Wietrzny	44,402
Gergory A. Nelson	30,577	Paul D. Amrozowicz	45,264
Keith M. Baxter	31,233	David M. Kettner	45,589
John D. Franzini	31,356	Adam J. Forman	46,707
Janine R. Novatt	32,592	Jay Chaskin	24,030
Jean C. Baker	35,433	Phyllis Y. Price	34,234
David G. Ryser	36,407	Henry J. Policinski	26,621
Bennett J. Berson	37,094	Ronald E. Myrick	26,315
Bruce T. Neel	37,406	Christian G. Cabou	35,467
Michael A. Jaskolski	37,551	Michael A. Della Penna	45,697

[ ] Additional attorney(s) and/or agents named on a supplemental priority sheet attached hereto

Please direct all correspondence to [ ] Customer Number or label _____ OR [X] Fill in correspondence

Name	Barry E. Sammons				
Address	Quarles & Brady				
Address	411 East Wisconsin Ave. Suite 2550				
City	Milwaukee	State	WI	Zip	53202-4497
Country	USA	Telephone	(414) 277-5000	Fax	(414) 271-3552

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Name of Sole or First Inventor: [ ] A petition has been filed for this unsigned inventor

Given	Josef	Middle	P.	Family	Debbins	Suffix	

Inventor's Signature _____ Date _____

Residence:	Waukesha	State	WI	Country	USA	Citizenship	US

Post Office	116 West Wabash Avenue
Post Office	

City	Waukesha	State	WI	Zip	53186	Country	US	Applicant Authority	

[X] Additional inventors are being named on supplemental sheet(s) attached hereto

DRAFT 02/22/99

Please type a plus sign (+) inside this box [+]

DECLARATION

ADDITIONAL INVENTOR(S)
Supplemental Sheet

Name of Additional Joint Inventor, if any: ☐ A petition has been filed for this unsigned inventor

Given	Middle	Family	Suffix
Kristine	L.	Gould	

Inventor's _____ Date _____

Residence:	State	Country	Citizenship
Delafield	WI	USA	US

Post Office: N8-W31314 Salem Court

Post Office:

City	State	Zip	Country	Applicant Authority
Delafield	WI	53108	USA	

Name of Additional Joint Inventor, if any: ☐ A petition has been filed for this unsigned inventor

Given	Middle Initial	Family Name	Suffix
Paul	E.	Licato	

Inventor's _____ Date _____

Residence:	State	Country	Citizenship
Wauwatosa	WI	USA	US

Post Office: 2602 North 82st Street

Post Office:

City	State	Zip	Country	Applicant Authority
Wauwatosa	WI	53213	USA	

Name of Additional Joint Inventor, if any: ☐ A petition has been filed for this unsigned inventor

Given	Middle	Family	Suffix
Jason	A.	Polzin	

Inventor's _____ Date _____

Residence:	State	Country	Citizenship
Lake Mills	WI	USA	US

Post Office: 140 Franklin Street

Post Office:

City	State	Zip	Country	Applicant Authority
Lake Mills	WI	53551	USA	

Name of Additional Joint Inventor, if any: ☐ A petition has been filed for this unsigned inventor

Given	Middle	Family	Suffix
Deepa		Thomas	

Inventor's _____ Date _____

Residence	State	Country	Citizenship
Waukesha	WI	USA	India

Post Office: 118 Manchester Drive

Post Office:

City	State	Zip	Country	Applicant Authority
Waukesha	WI	53188	USA	

☐ ☐ Additional inventors are being named on supplemental sheet(s) attached hereto

PTO/SB/01 (08-03)